

EXAMINER'S AMENDMENT

1. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview with Mr. Geringson (reg. 57,033) on 2/27/2009.

The application has been amended as follows:

1. (Currently Amended) A system for designing and executing processes, said system comprising:

an introspection module in a generic introspection framework that automatically generates a catalog of generic components by introspecting a set of exposed application programming interfaces (APIs) of a plurality of heterogeneous applications created in different programming languages and transforming a plurality of implementation-specific components of said heterogeneous applications into the generic components of said catalog by binding identifiers of the generic components with identifiers of the implementation-specific components wherein the catalog contains a series of entries in a binding table automatically created during the introspection of the APIs, each entry representing a generic component which, when invoked, is bound to at least one of the implementation-specific components of said heterogeneous applications upon execution of the processes, the catalog further containing a module comprising the generic components associated with a folder comprising an XML file when the implementation-specific components are saved in the catalog as the generic components;

an implementation module that includes procedures to access the APIs and maps the implementation-specific components to the generic components by using its own introspection mechanism and sends the mapped implementation-specific components to the generic introspection framework;

a component manager coupled to the introspection module and ~~operable~~ configured to manage said catalog automatically generated by the introspection module by defining and organizing the generic components in said catalog; and

a process designer coupled to the component manager and ~~operable~~ configured to:

select at least one of the generic components from said catalog managed by the component manager; and

graphically construct a process definition for the processes that includes a series of graphically represented activities linked by one or more transitions and associated with abstract roles specified for the activities wherein at least one activity of said process definition invokes the selected generic component from said catalog;

whereby a single process is assembled from the implementation-specific existing components of ~~multiple~~ the different programming languages, the implementation-specific components being compiled modular routines of the heterogeneous applications that have been previously created in different programming languages; and

a repository for storing the process definition; and

one or more process engines that execute said process definition to instantiate a process instance, wherein the process instance interacts with the plurality of heterogeneous applications by invoking the generic components in said catalog and wherein the process instance integrates the plurality of heterogeneous applications into the single process by invoking services from the plurality of heterogeneous applications during execution of the activities of said processes;

wherein during execution of the process instance, the ~~multiple~~ compiled modular routines of the different heterogeneous applications are invoked in order to complete the activities of said process instance.

3 – 4. (Canceled)

5. (Currently Amended) The system of Claim 1, further comprising a debugger coupled to the process designer and ~~operable~~ configured to detect an error of the process definition.

6. (Currently Amended) The system of Claim 1, further comprising:
a data warehouse coupled to the one or more process engines and ~~operable~~ configured to store transactional data describing the ~~executed~~ process instance; and
a data server coupled to the data ~~operable~~ configured to organize the transactional data.

21. (Currently Amended) The system of Claim 1 wherein said ~~introspection module further includes at least one~~ implementation module ~~that~~ is used to access implementation-specific components associated with at least one of: Java JAVA, Standard Query Language (SQL), Automation, Enterprise JavaBeans (EJB), CORBA, Remote Method Invocation (RMI), Extensible Markup Language (XML) schemas, Web Services and Java JAVA Naming and Directory Interface (JNDI).

22. (Canceled)

23. (Currently Amended) A computer implemented method for designing and executing processes, said method comprising:

automatically generating a catalog of generic components in a generic introspection framework by introspecting a set of exposed application programming interfaces (APIs) of a plurality of heterogeneous applications implemented in multiple different programming languages and ~~translating~~ transforming implementation-specific components of the plurality of heterogeneous applications into the generic components of said catalog by binding identifiers of the generic components with identifiers of the implementation-specific components, wherein the catalog contains a series of entries in a binding table automatically created during the introspection of the APIs, each entry representing a generic component which, when invoked, is bound to at least one of the implementation-specific components of said heterogeneous applications upon execution of the processes, the catalog further containing a module comprising the generic components associated with a folder comprising an XML file when the implementation-specific components are saved in the catalog as the generic components;

providing an implementation module that includes procedures to access the APIs and maps the implementation-specific components to the generic components by using its own introspection mechanism and sends the mapped implementation-specific components to the generic introspection framework;

selecting at least one of the generic components from the catalog;

graphically constructing one or more processes definitions for the processes, each process definition including a series of graphically represented activities linked by one or more transitions and associated with abstract roles specified for the activities wherein at least one activity of said processes invokes the at least one generic component selected from said catalog, whereby a single process is assembled from existing the implementation-specific components of the multiple different programming languages, the implementation-specific components being compiled modular routines of the heterogeneous applications that have been previously created in the multiple different programming languages; and

executing the ~~generated~~ constructed one or more process definitions at one or more process engines in order to instantiate a process instance, wherein the process instance interacts with the plurality of heterogeneous applications by invoking the generic components of said catalog and wherein the process instance integrates the plurality of heterogeneous applications into the single process by invoking services from the plurality of heterogeneous applications during execution of the activities of said processes.

25. (Canceled)

27. (Currently Amended) The method of Claim 23, further comprising:

storing transactional data describing the ~~executed~~ process instance in a data warehouse; and organizing the transactional data at a data server.

28. (Canceled)

29. (Currently Amended) The method of Claim 23 wherein ~~introspecting further includes providing at least one~~ the implementation module ~~that~~ is used to access implementation-specific components associated with at least one of: ~~Java~~ JAVA, Standard Query Language (SQL), Automation, Enterprise JavaBeans (EJB), CORBA, Remote Method Invocation (RMI), Extensible Markup Language (XML) schemas, Web Services and ~~Java~~ JAVA Naming and Directory Interface (JNDI).

30. (Currently Amended) A computer having one or more processors to design and execute processes, wherein instructions executed by the one or more processors cause the computer to perform the steps of: ~~readable medium having instructions stored thereon which when executed by one or more processors cause a system to:~~

automatically generating a catalog of generic components in a generic introspection framework by introspecting a set of exposed application programming interfaces (APIs) of a plurality of heterogeneous applications implemented in multiple different programming languages and ~~translating~~ transforming implementation-specific components of the plurality of heterogeneous applications into the generic components of said catalog by binding identifiers of the generic components with identifiers of the implementation-specific components, wherein the catalog contains a series of entries in a binding table automatically created during the introspection of the APIs, each entry representing a generic component which, when invoked, is bound to at least one of the implementation-specific components of said heterogeneous applications upon execution of the processes, the catalog further containing a module comprising the generic components associated with a folder comprising an XML file when the implementation-specific components are saved in the catalog as the generic components;

providing an implementation module that includes procedures to access the APIs and maps the implementation-specific components to the generic components by using its own introspection mechanism and sends the mapped implementation-specific components to the generic introspection framework;

selecting at least one of the generic components from the catalog;

graphically constructing one or more processes definitions for the processes, each process definition including a series of graphically represented activities linked by one or more transitions and associated with abstract roles specified for the activities wherein at least one activity of said processes invokes the at least one generic component selected from said catalog, whereby a single process is assembled from ~~existing~~ the implementation-specific components of the multiple different programming languages, the implementation-specific components being compiled modular routines of the heterogeneous applications that have been previously created in the multiple different programming languages; and

executing the ~~generated~~ constructed one or more process definitions at one or more process engines in order to instantiate a process instance, wherein the process instance interacts with the plurality of heterogeneous applications by invoking the generic components of said catalog and wherein the process instance integrates the plurality of heterogeneous applications into the single process by invoking services from the plurality of heterogeneous applications during execution of the activities of said processes.

31. (Currently Amended) The system of claim 1 wherein said process definition is published to the repository before being deployed to the one or more process ~~engine engines~~.

32. (Currently Amended) The system of claim 1 wherein the ~~catalog contains one or more entries, each entry including~~ entries of the catalog include metadata that describes at least one of the plurality of implementation-specific components.

33. (Currently Amended) The system of claim 1 wherein an activity of said process definition connects to a subprocess ~~that operates as a process~~.

35. (Canceled)

37. (Currently Amended) The system of claim [[37]]36, wherein the measures of said cube are filtered by the dimensions of said cube in order to generate an analytical representation.

38. (Currently Amended) The system of claim 37 wherein the measures are identified with one or more of the following: a number of process instances in [[an]] activities, the number of instances completed by said single process, execution time of a process instance and an average wait time for an activity.

Examiner's Statement of Reason(s) for Allowance

2. Claims 1, 2, 5, 6, 21, 23, 24, 26, 27, 29-34, and 36-40 (renumbered as 1-20) are allowed.
3. The following is an examiner's statement of reasons for allowance:

The cited prior arts of record, i.e. Sarkar, Kilgore, Flores, Chinnici, Ims, Acker, Ambrose, Monday, taken alone or in combination, fail to teach or fairly suggest at least: an introspection module that automatically generates a catalog of generic components...by binding...in a binding table...containing a module comprising the generic components associated with a folder comprising an XML file when the implementation-specific components are saved in the catalog as the generic components; an implementation module that includes procedures to access the APIs and maps the implementation-specific components to the generic components by using its own introspection mechanism and sends the mapped implementation-specific components to the generic introspection framework; graphically construct a process definition for the processes... associated with abstract roles specified for the activities wherein at least one activity of said process definition invokes the selected generic component from said catalog as recited in the independent claims 1, 23, and 30.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

4. Any inquiry concerning this communication or earlier communications from the examiner should be directed to INSUN KANG whose telephone number is (571)272-3724. The examiner can normally be reached on M-R 7:30-6 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Lewis A. Bullock, Jr. can be reached on 571-272-3759. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Insun Kang/
Examiner, Art Unit 2193